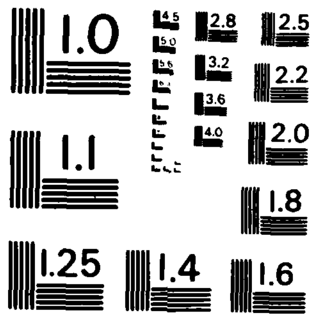
AD-A145 737 UNITED STATES AIR FORCE ON RESEARCH ON ALGEBRAIC MANIPULATION(U) MASSACHUSETTS INST OF TECH CAMBRIDGE DEPT OF ELECTRICAL ENGINEERING J MOSES AUG 84 1/1

UNCLASSIFIED AFOSR-TR-84-0785 AFOSR-80-0250 F/G 12/1 NL

END
DATE
FILMED
10 84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

FINAL REPORT

TO THE

UNITED STATES AIR FORCE

ON

RESEARCH ON ALGEBRAIC MANIPULATION

GRANT AFOSR-80-0250

by

Joel Moses
Massachusetts Institute of Technology
August, 1984

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 84-0785
6a. NAME OF PERFORMING ORGANIZATION Massachusetts Institute of Technology — 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research
6c. ADDRESS (City, State and ZIP Code) Department of Electrical Engineering and Computer Science, Cambridge MA 02139	7b. ADDRESS (City, State and ZIP Code) Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR — 8b. OFFICE SYMBOL (If applicable) NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-80-0250
8c. ADDRESS (City, State and ZIP Code) Bolling AFB DC 20332	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. 61102F; PROJECT NO. 2304; TASK NO. A4; WORK UNIT NO.

11. TITLE (Include Security Classification)

RESEARCH ON ALGEBRAIC MANIPULATION

12. PERSONAL AUTHOR(S)

Joel Moses

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Final	FROM 1/7/83 TO 30/6/84	AUG 84	12

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

During this period the single investigator refined his technique for symbolic integration, implemented it in a computer program in MACSYMA, and produced and presented a paper describing his achievement entitled, "An experiment toward a general quadrature for second order linear ordinary differential equations by symbolic computation." A measure of his success is that he was able to integrate successfully 90% of the 542 equations in Kamke's famous table. (Since 50 of these equations involved arbitrary functions, etc., for which the program was not designed, the success rate is more appropriately 96%.)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Robert N. Buchal	22b. TELEPHONE NUMBER (Include Area Code) (202) 767-4939 — 22c. OFFICE SYMBOL NM

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

84 09 17 042

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

The bulk of the research performed in the past two years on this contract was on the solution in closed form of second order ODEs. This was done largely by Professor Shunro Watanabe while he visited us from Japan. A paper on this work was presented on July, 1984 in Cambridge, England and is attached.

As the paper indicates, Watanabe's approach, which is based on transforming most equations into a variant of Riemannian functions, is very successful. It solves over 90% of all second order equations in Kamke's famous book. If one eliminates differential equations with general coefficients (e.g., f(x)), then it solves over 96% of the equations. Watanabe's paper explains the types of problems that remained unsolved.

We should note that Watanabe's program is more general than Kamke's book. It is now available as a program in MACSYMA.

We are very pleased with our Air Force support over the years. With this support we were able to complete a PhD thesis by Zippel on the GCD algorithm. This pathbreaking thesis provided a probabilistic algorithm that is the best general purpose GCD algorithm. Barry Trager has almost completed his PhD thesis on algebraic integration. This thesis presents a very efficient algorithm using much machinery from algebraic geometry. When it is completed this fall, we expect to submit it to the Air Force as well. Finally, we were able to sponsor Prof. Watanabe's work. On the whole, our association with AFOSR has been outstanding. We hope to continue it at some future point.

AN EXPERIMENT TOWARD A GENERAL QUADRATURE FOR
SECOND ORDER LINEAR ORDINARY DIFFERENTIAL EQUATIONS
BY SYMBOLIC COMPUTATION

Shunro Watanabe
Department of Mathematics, Tsuda College
Kodaira, Tokyo 187, Japan

1. Why experiment?

The second order linear ordinary differential equations (L ODE) is the most important class in ODE. The classical mathematical theories for L ODE had developed in 19th and early 20th centuries. Many mathematicians made the theories and methods to find and solve liouvillian or algebraic solutions for L ODE. However it seems to us they did not offer any general procedure that can solve these equations. ([1])

On the other hand, during the last 15 years many people tried to write programs that can solve the equations in L ODE by Symbolic Computation. For example, J.Golden E.Lafferty and others wrote an solver for ODE on MACSYMA, called ODE, which is a collection of algorithms including Y.Avgoustis' simplification program for hypergeometric equations and P.Schmidt's solver for Riccati's equations with coefficients in Q(x), rational functions of x. ([2], [3])

Recently two papers appeared. They offered general algorithms for these equations. J.Kovacic's algorithm can find and solve all the liouvillian and algebraic solutions for second order L ODE with coefficients in C(x). B.Saunders implemented Kovacic's algorithm. ([4]) M.Singer's algorithm can find and solve all the liouvillian and algebraic solutions for the n-th order L ODE with coefficients in F, a finite algebraic extension of Q(x). ([5])

Even after the appearance of these two papers, if one wants to implement a solver for a large class of equations, the following direction seems to be still valuable: "Given a differential equation whose form or structure is not immediately recognizable, one looks for transformations which will convert the given problem into one which is known." ([6]) In this paper, I shall show an experiment toward a general quadrature for second order L ODE with coefficients in elementary functions.

I wrote a program within the classical knowledge on ODE. ([1],[8],[9]) It consists of some 1400 lines by MACSYMA language and I tested this program on PDP-10 using 542 equations in Kamke's table. In these 542 equations we can use 492 equations as meaningful test data. ([7]) Our program solved 473 equations. It means our solvable rate is more than 96%. The computaion times are almost between 10 seconds and 60 seconds. In this experiment, I found an essential error (2-291th equation)

Proc. EUROSAM '84, Springer Lec. Notes Comp. Sci. 174.

and other errors (2-125c(c) and 2-187a) in Kamke's table. Also our program solved a few equations which are essentially different equations from those in Kamke's table. I printed all the processes of calculations for the 473 equations and others.

2. The strategy for solving.

Our approach for solving Kamke's equations is to find a proper transformation of variables which will convert a given equation to a more simple equation. Usually it is very difficult to determine which equation is more simple. However we can guess as follows : if the coefficients of an equation have $\exp(x^2)$ and the coefficients of another equation have only $\exp(x)$, the latter equation must be more simple than the former equation. When all the coefficients of an equation are rational functions of x we may think that the degree of the difficulties for solving increases as the number or the ranks of the singular points increase. Thus we had rough criterions for simplicity of equations.

Then how can we find proper transformations? I used only one technique for our program. First we will recognize the pattern for the given equation. Here I mean the pattern not only as external form but also as a kind of characterization using the informations obtained by calculation. Then we will get several candidate transformations that have a few undetermined parameters. We will try to determine these parameters by applying the transformations to a given equation. Therefore we used the following strategy for our program.

step 1. If the equation contains elementary transcendental functions and if the arguments in the deepest parts of it have a common rational function $k(x)$ that is not x then we try to remove $k(x)$ by the transformation $t=k(x)$. If we success then go to step 5, if we fail then go to step 4.

step 2. If the equation contains elementary transcendental functions and if all the arguments of these functions are x then we try to remove these functions by the transformation $t=e(x)$, where $e(x)$ is one of the transcendental functions.
If we success then go to step 5, if we fail then go to step 4.

step 3. If all the coefficients of the equation are rational functions of x and parameters then we count all the singular points and calculate their ranks. If the equation has only three regular singular points or it has one regular singular point and one irregular singular point of rank one or it is the easily solvable equation then we solve it using theories. If the equation is a prototype then we say so. If we success then go to step 6.

step 4. We try to find the proper transformations of the form

$u=f(x)y$, $u=y'$, or $t=g(x)$

where $f(x)$ and $g(x)$ are elementary or algebraic function of x. Often $f(x)$ and $g(x)$ have undetermined parameters, and we must determine them so as the transformation can simplify the equation. If we fail we cannot solve it.

step 5. We store this successful transformation of variable to the top of a stack. We replace the new variables u or t in the transformed equation by y or x and we use it as new equation. Go to step 1.

step 6. We calculate the solution of the first equation from the series of transformations on the stack and the solution of the last equation.

When we wrote our program according to the above strategy, we used the following loose principles : 1) We should prepare enough transformations for solving our equations. But it is better to use pattern matchings in small numbers. 2) We should use back-tracking technique only under the restricted condition. At least the number of trials in an environment must be small.

3. Details on the transformations.

Let us consider step 2 in our strategy. When we find trigonometric functions for a given equation, we try to remove these functions from it using $t=\sin(x)$ or $t=\cos(x)$. When one transformation succeeded and another transformation failed, we can use the succeeded one. When both of them succeeded, we must select the one which will bring us more simple equation. When both of them failed, we cannot remove trigonometric functions from it.

When we find hyperbolic functions for a given equation, we try to remove these functions from it using $t=\sinh(x)$ or $t=\cosh(x)$. We can determine which transformation is proper or not using the same procedure as the case of trigonometric functions. When we find exponential or logarithmic functions for a given equation, we try to remove them from it using $t=e^x$ or $t=\log(x)$ or $t=x(\log(x)-1)$.

Now let us consider step 4 in our strategy. First we try to simplify it using $t=x^r$. For this purpose we try to rewrite our equation to the form $x^2y''+xf(x^r)y'+g(x^r)y=0$. Where r is an undetermined parameter. When r is 2 or 3, or -1 or 1/2, it is not so difficult to determine r. But when r is b or -b or b+1, where b is an another symbol, it is not so easy to determine r.

Then we try to simplify it using $y=\exp(ax^r)u$, where a and r are two undetermined parameters. By this transformation we can expect two directions for simplification. One is to reduce the rank of the irregular singularity, and another is to transform our equation to easily solvable equation as $y''+f(x)y'=0$. To reduce the rank we can use the value of rank as r. But to transform our equation to $y''+f(x)y'=0$ we must look for the value of r around the value of rank. Sometimes we go through this step two or three times. Then we must determine the value r under the condition that the value of the successor must be less than the value of the predecessor.

In this case we have one difficulty. The undetermined parameter 'a' in $\exp(ax^r)$ satisfies a quadratic equation. So we have two values for candidate. The two transformed equations corresponding to these values have often same simplicity. Therefore the first version of our program asks for us which value is preferable. Of course it is for the memory limitation's sake.

After this transformation, we still try to simplify our equation using $y=(x-a)^k u$, where a and k are undetermined parameters. By this transformation we can expect two directions for simplification. One is to remove an apparent singular point from the equation. For this purpose we must select an apparent singular point as 'a' and one of the characteristic roots as k. It is not necessary to decide whether a singular point is apparent or not, because the possible number of a and k is finite.

Another direction is to transform the equation to $y''+f(x)y'=0$. For this purpose it is not necessary to select a singular point for a. These processes are a kind of pattern matchings and their applications for transformations. Then we try to use more explicit patterns.

4. What are our patterns?

In our problem a data or an equation corresponds to a program which can solve the equation. Now we have 542 relevant equations in Kamke's table. Therefore if I wrote 542 programs, then the collection of these programs is a solver for Kamke's equations. However it is too big to be a practical solver. Then we try to find similar parts in this huge program and try to reduce its size by replacing those similar parts by subroutines. These subroutines correspond to patterns.

For example a few equations in Kamke's table pass through similar route in step 4, then we can use a proper pattern to save calculation time. The equations 2-54 and 2-55 in Kamke's table are such examples. Let us consider the equation 2-189 as next example. It is transformed to Bessel's equation (2-162). Our program can solve it easily. However when we solve all of the 542 equation we will meet them 54 times. Therefore we added the pattern 2-189 to our program to save computation time.

In a practical sense how can we find a pattern? Let us consider the easiest example, equation 2-442. It has the form $f(x)y''+xy'-y=0$. When the equation 2-419 is given to us, let us look at it. It has the form : $x^2y''\cos(x)+(x^2\sin(x)-2x\cos(x))y'+(2\cos(x)-x\sin(x))y=0$. After we divided the both sides by $-(2\cos(x)-x\sin(x))$ we can get $f(x)=x^2/(x\sin(x)-2\cos(x))$. The pattern 2-442 has a special solution x, so we can easily solve it.

Then is it always possible to determine whether a pattern matches to an equation or not? The equation 2-77a has the form : $y''+(f+g)y'+(f'+fg)y=0$, where f and g are arbitrary functions of x. When we tried to match this pattern to $y''+py'+qy=0$, we will see that f must be the solution of a Riccati's equation : $f'+pf-f^2-q=0$. But it is very difficult to solve this equation, it is equivalent to our problem.

5. Examples.

Example 1. The following are almost raw print-out for the 2-344th equation.

```
(C3) showtime:true$
Time= 5 msec.

(C4) /* September 10, 1983 */

loadfile(pmain,fasl);

PMAIN FASL DSK SWATAN being loaded
Loading done
Time= 333 msec.
(D4)                                DONE

(C5) batch(exampl,test);

(C6) /* 2-344 */

K344:X^4*'DIFF(Y,X,2)+(EXP(2/X)-V^2)*Y=0;
Time= 72 msec.
                          2
                        4 d Y       2/X     2
(D6)                   X  --- + (%E     - V ) Y = 0
                            2
                          dX

(C7) /* see 2-182(24) */


Time= 340 msec.
(D7)                                BATCH DONE

(C8) lode2(k344,0);

           2        2/X    2
          d Y    (%E    - V ) Y
we solve  --- + -------------- = 0
            2           4
          dX           X

               1
we use T = -
               X
                            dY
                    2    2 --
                   d Y      dT       2 T    2
the result is  --- + ---- + (%E     - V ) Y = 0
                     2    T
                   dT
                      dY
              2    2 --
             d Y      dX       2 X    2
we solve  --- + ---- + (%E     - V ) Y = 0
               2    X
             dX

SOLVE FASL DSK MACSYM being loaded
Loading done
we use T = 2 X
                            dY
                    2    2 --      2     T
                   d Y      dT   (V  - %E ) Y
the result is  --- + ---- - -------------- = 0
                     2    T           4
                   dT
                      dY
              2    2 --       X    2
             d Y      dX   (%E  - V ) Y
we solve  --- + ---- + -------------- = 0
               2    X           4
             dX
```

```
               U
we use Y = -
               X
                              X    2       2
                         U (%E  - V )   d U
the result is  ------------ + --- = 0
                              4             2
                                          dX
              2          X    2
             d Y    (%E  - V ) Y
we solve  --- + ------------ = 0
               2          4
             dX
                    X
we use T = %E
                            dY
                    2    --       2
                   d Y   dT   (V  - T) Y
the result is  --- + -- - ---------- = 0
                     2   T          2
                   dT             4 T
                      dY
              2    --         2
             d Y   dX   (X - V ) Y
we solve  --- + -- + ---------- = 0
               2   X          2
             dX             4 X

         [ INF  ASTERISK    0  ]
         [                     ]
         [                  V  ]
         [  0       - %I   - - ]
y=P [                   2  ] (x)
         [                     ]
         [                  V  ]
         [  0         %I    -  ]
         [                  2  ]

y= Y            (SQRT(X))
     B, ABS(V)
the solution of the last eq. is Y            (SQRT(X))
                                       B, ABS(V)
the solution of the first eq. is
Time= 24239 msec.
                                                  1/X
                                  X Y           (%E    )
                                     B, ABS(V)
(D8)                          ----------------------
                                             2
```

In the above example $y_{B,n}(x)$ is the general solution of the Bessel's equation : $x^2y''+xy'+(x^2-n^2)y=0$.

Example 2. Print-out for the 2-378a equation in Kamke's table.

```
(C6) /* 378A,522 */

K522:X*(X-1)*(X+1)^2*'DIFF(Y,X,2)+2*X*(X+1)*(X-3)*'DIFF(Y,X)-2*(X-1)*Y=0;
Time= 56 msec.
```

$$\text{(D6)}\quad (X-1)\,X\,(X+1)^2\,\frac{d^2Y}{dX^2} + 2\,(X-3)\,X\,(X+1)\,\frac{dY}{dX} - 2\,(X-1)\,Y = 0$$

```
(C8) lode2(k522,0);
```

we solve $\dfrac{d^2Y}{dX^2} + \dfrac{(2X-6)\dfrac{dY}{dX}}{X^2-1} - \dfrac{2Y}{X^3+2X^2+X} = 0$

we use $Y = \dfrac{U}{(X+1)^2}$

the result is $\dfrac{2U}{X^2-X} - \dfrac{2\dfrac{dU}{dX}}{X-1} + \dfrac{d^2U}{dX^2} = 0$

we solve $\dfrac{d^2Y}{dX^2} - \dfrac{2\dfrac{dY}{dX}}{X-1} + \dfrac{2Y}{X^2-X} = 0$

it matched with k442

we use Y = U X

the result is $\dfrac{dU}{dX}\left(\dfrac{2}{X} - \dfrac{2}{X-1}\right) + \dfrac{d^2U}{dX^2} = 0$

we solve $\dfrac{d^2Y}{dX^2} - \dfrac{2\dfrac{dY}{dX}}{X^2-X} = 0$

the solution of the first eq. is

Time= 20291 msec.

$$\text{(D8)}\quad \frac{X\left(K1\left(-2\,\mathrm{LOG}(X) + X - \frac{1}{X}\right) + K2\right)}{(X+1)^2}$$

Example 3. Print-out for the 2-430 equation in Kamke's table.

```
(C6) K430:'DIFF(Y,X,2)*SIN(X)^2+'DIFF(Y,X,1)*SIN(X)*COS(X)
     +(V*(V+1)*SIN(X)^2-N^2)*Y=0;
Time= 93 msec.
```

$$\text{(D6)}\quad \mathrm{SIN}^2(X)\,\frac{d^2Y}{dX^2} + \mathrm{COS}(X)\,\mathrm{SIN}(X)\,\frac{dY}{dX} + \left(V\,(V+1)\,\mathrm{SIN}^2(X) - N^2\right)Y = 0$$

```
(C8) lode2(k430,0);
```

```
                                dY
          2      COS(X) --         2           2         2
         d Y                dX    ((V  + V) SIN (X) - N ) Y
we solve --- + --------- + ------------------------ = 0
           2      SIN(X)                      2
         dX                                SIN (X)

SOLVE FASL DSK MACSYM being loaded
Loading done
we use T = COS(X)
                              dY
                  2     2 T.--         2           2     2             2
                 d Y          dT    ((T  - 1) V  + (T  - 1) V + N ) Y
the result is --- + ------ - ---------------------------------- = 0
                    2     2                        4       2
                 dT     T  - 1                    T  - 2 T  + 1
                           dY
          2      2 X --         2         2     2            2
         d Y             dX    ((V  + V) X  - V  - V + N ) Y
we solve --- + ------ + --------------------------- = 0
           2     2                   4       2
         dX     X  - 1              X  - 2 X  + 1

it matched with  k372
                           ABS(N)
                           ------
                   2            2
we use Y = U (X  - 1)
                   2
                  d U   2                           dU            2
the result is --- (X  - 1) + 2 (ABS(N) + 1) -- X  + U (- V  - V + ABS(N) (ABS(N) + 1)) = 0
                    2                               dX
                  dX
                                           dY
          2      (2 ABS(N) + 2) X --         2                      2
         d Y                              dX    (V  + V - ABS(N) - N ) Y
we solve --- + -------------------- - ------------------------ = 0
           2           2                              2
         dX           X  - 1                         X  - 1

 the type is hypergeometric
 the solution may be written by Riemann's P-functions as follows
    [     1          - 1            INF         ]
    [                                           ]
y=P [ - ABS(N)   - ABS(N)     ABS(N) - V       ] (x)
    [                                           ]
    [     0           0        V + ABS(N) + 1 ]
do you replace in ABS(N) - V ? type y or n
n;
is - 2 V - 1 an odd integer? type y or n
n;
is 2 V + 1 an odd integer? type y or n
n;
Is ABS(N) a positive integer? type y or n
y;
Is 2 V + 1 a positive integer? type y or n
n;
Is - 2 V - 1 a positive integer? type y or n
n;
    ABS(N)
   d
y= -------- (Y (V, X))
    ABS(N)     L
   dX
where Y (V, X)
       L          is the solution of Legendre's eq:(x^2-1)*y''+2*x*y'-v1*(v1+1)*y=0
the solution of the first eq. is
Time= 29504 msec.
                                                                        ABS(N)
                       ABS(N)                                           ------
                      d                                     2              2
(D7)          (-------------- (Y (V, COS(X)))) (COS (X) - 1)
                         ABS(N)    L
                dCOS(X)
```

6. The result of our experiment.

There are 542 second order L ODE in Kamke's table. In these equations we have 39 equations which contain arbitrary functions and 11 equations which contain non elementary transcendental functions. Our program solved 473 equations out of relevant 492 equations. The rate of solved equation is more than 96%. Our program solved 488 equations out of all the 542 equations. The rate of solved equation without any restriction is more than 90%.

When will we say "We could solve it." or "We could not solve it."? When the most simplified equation *is proto-type or has a solution that is representable by* elementary functions or algebraic functions, the equation was solved.

type	classes \ the number of	solved equat.	unsolved equation	total	the type of last tr.eq.			
					s0	s1	s2	s3
s 0	constant coefficients or first order equation of y'	18	0	18				
s 1	Riemann's equation of confluent type	114	0	114	5	109		
s 2	Riemann's equation	99	1	100	13		86	
s 3	$t=x^3 \rightarrow$ s1 or s2	118	0	118	26	66	26	
s 4	coefficients contain exponential functions	15	3	18	4	10	1	
s 5	coefficients contain logarithmic functions	4	2	6	4			
s 6	coefficients contain trigonometric functions	55	2	57	11	13	29	2
s 7	coefficients contain hyperbolic functions	7	0	7			6	1
s 8	other equations with coefficients in Q(x)	43	11	54	19	5	9	10
	sub total	473	19	492	100	203	157	13
s 9	coefficients contain transcendental functions	2	9	11				
s10	coefficients contain any functions of x	13	26	39				
	sub total	15	35	50				
	total	485	57	542				

Table 1.

last equation		number of solutions representable by									
					the solution of the equation of						
last class	number of eq	elemen func	algeb func	ellip func	Kummer	Bessel	Whitta -ker	Legen -dre	Gauss	Mattieu	Others
s0	100	100									
s1	203	87			6	51	59				
s2	157	68	9	3				36	41		
s8	13									6	7
total	473	255									

Table 2.

pattern	transfor-mation	frequ-ency	pattern	transfor-mation	frequ-ency	pattern	transfor-mation	frequ-ency
2- 41		2	2-367	$y=(x^2+1)^{\frac{m}{2}}u$	1	2-218a		1
2- 54	$y=\exp(ax)u$	2	2-372	$y=(x^2-1)^{\frac{m}{2}}u$	11		$y=(x-a)^r u$	9
2- 55	$y=\exp(ax^2)u$	5	2-339	$t=\sqrt{x}$	5		$y=\exp(ax^r)u$	29
2- 78	$y=u/(x^2-1)$	3	2-394	$t=c\log(\frac{x}{ax+b})$	1		$\begin{cases} y=u/\sin(x) \\ y=u/\cos(x) \end{cases}$	8
2-120	(to Whittaker)	39	2-442	$y=(x-a)u$	28		$y=\log(x)u$	1
2-130	$t=\sqrt{x}$	2	2-188a	(prototype)	1		$t=\frac{ax+b}{cx+d}$	31
2-189	(to Bessel)	54	2-231a	$t=\operatorname{asinh}(x)$	1		$t=x^r$	137
2-248	(proto-type)	3	2-wit	(to Whittaker)	28		$\begin{cases} t=\sin(x) \\ t=\cos(x) \end{cases}$	44
2-269	$y=x^r u$	1	2- 79		1		$t=e^x$	11
2-297	$t=\operatorname{asinh}(\sqrt{a}x)/\sqrt{a}$	4	2-128	$y=u/x$	1		$\begin{cases} t=\sinh(x) \\ t=\cosh(x) \end{cases}$	7
2-357	$t=\sqrt{x^2+1}$	5	2-220		2		$t=\log(x)$	1
2-359	$t=1/x$	2	2-221		1		$t=x(\log(x)-1)$	1
2-363	$t=\frac{1}{2}(x+\frac{1}{x})$	4	2-76a		1		$y=\cos(x)u$	1

$$\text{2-wit}: \; x^2y''-x(2a+2bx)y'+(a(a+1)+(\frac{1}{4}-m^2)+2ab+pk)x+(b^2-\frac{p^2}{4})x^2)y=0.$$

Table 3.

In table 3 we can read how many times a pattern matched to its equations or how many times a transformation was done in our experiment. For example a pattern 2-wit which we cannot find in Kamke's table matched to 28 equations, and $t=\sin(x)$ or $t=\cos(x)$ was done 44 times in our experiment.

equation	reason for unsolved	equation	reason for unsolved	equation	reason for unsolved
2- 15	not implemented	2-330	too general	2-427	too special
2- 19	not implemented	2-341	not implemented	2-23a	too difficult
2-127	too special	2-362	not implemented	2-115b	too difficult
2-216	not implemented	2-364	not implemented	2-115c	too difficult
2-261	is not well-known	2-399	not implemented	2-354b	too general
2-267	is not well-known	2-407	too general		
2-283	too special	2-408	not implemented		

Table 4. The list of all the unsolved equations in s1-s8.

7. References.

1. A.Forsyth,"Theory of Differential Equations vol.IV", Dover,(1960).
2. E.Lafferty,"Hypergeometric Reduction-An Adventure in Pattern Matching", Proc. 1979 MACSYMA User Conf, pp.465-481.
3. Y.Avgoustis,"Symbolic Laplace Transforms of Special Functions", Proc. 1979 MACSYMA User Conf. pp.21-40.
4. B.Saunders,"An Implementation of Kovacic's algorithm for solving second order linear homogeneous differential equations", Proc. ACM Symp. SYMSAC'81, pp.105-108.
5. M.Singer,"Liouvillian Solutions of n-th Order Homogeneous Linear Differential Equations", Amer.J.Math. vol.103,no.4, 1980, pp.661-682.
6. R.Pavelle, M.Rothstein, J.Fitch,"Computer Algebra", Scientific American, Dec. 1981, pp.136-146,151-152.
7. E.Kamke,"Differential Gleichungen-Lösungsmethoden und Lösungen", Chelsea,(1959).
8. J.Della Dora, E.Tournier,"Formal Solutions of Differntial Equations in the Neighbourhood of Singular Points", Proc. ACM Symp. SYMSAC'81, pp.25-29.
9. S.Watanabe,"A Technique for Solving Ordinary Differential Equations Using Riemann's P-functions", Proc. ACM Symp. SYMSAC'81, pp.36-43.

8. Acknowledgements.

The work described in this paper was performed with MACSYMA which is supported by the U.S.Air Force under grant F49620-79-020. I am very grateful to J.Moses and the member of Mathlab group in MIT. I could not write my program without the help of J.Golden, E.Golden, and R.Pavelle during the period 4/1/82-9/30/83.

EN

DAT